

# 东南大学电工电子实验中心

## 实 验 报 告

课程名称： 数字电路实验

### 第 4 次实验

实验名称： FPGA 时序逻辑设计

院（系）： 电气工程学院 专 业： 电气工程及其自动化

姓 名： 王皓冬 学 号： 16022627

实 验 室： 401 实验组别：

同组人员：  实验时间：

评定成绩：  审阅教师：

## 一、实验目的

- 1、进一步学习时序逻辑电路的分析和设计方法，掌握状态机设计方法
- 2、了解可编程数字系统设计的流程，掌握 Quartus II 软件的使用方法
- 3、掌握原理图输入方式设计时序逻辑电路的方法和流程

## 二、实验原理

### 0 输入信号与输出信号编码

#### 输入信号

进行动态验证时，由于该电路以预置的形式产生了信号，而未采用 input 形式，因此该实验中**无输入信号**。而需要手动加载数据时，采用 CLK 按键与拨码开关与逻辑的方式进行信号输入。故

**Input:** 手动加载的序列信号，用拨码开关实现。

为了同一两种检测方式，引入选择功能，采用拨码开关实现。即

**Model:** 模式选择。为“1”时手动输入信号进行验证，为“0”时检测预置的信号“1011011100100”。

#### 输出信号

**OD:** Order，输出预定信号“111001”部分模块的实际输出。

**Y:** 输出的待测信号“1011011011100100”的实际输出。

**Z:** 代表“待测信号中是否被检测到含有预定信号”，“0”代表未检测到，“1”代表检测到。

**CLKo:** 输出时钟信号，用于拓展功能，便于验证。o 代表 out。

**Modelo:** 输出 Model，直观显示当前检测模式。用于拓展功能，便于验证。o 代表 out。

**X:** 对每一个 CLK 内的 OD 与 Y 异或，用于表示 OD 与 Y 信号是否相等。“0”代表不相等，“1”代表相等。用于拓展功能，便于验证。

#### 时钟信号

用 **CLK** 代表时钟信号。实验中的时钟信号是唯一的，仿真时，CLK 的周期为 20ns。由于利用 FPGA 单步验证时，其按钮按下产生低电平，与通常的 CLK 信号相反，因此电路设计时采用了 CLKi(i 代表 in)作为输入时钟信号的标识，CLK 则仍作为接入各芯片的时钟信号标识。

#### 其他标识说明

**LD:** 二进制码实现序列检测器时接入 161LD 端的导线逻辑。LD 为“0”时，161 实现加载数据的功能。

**Nand:** One-Hot 码中接回 D<sub>0</sub> 触发器、用作自启动的导线逻辑。Nand 为“1”，即与非的项中含有逻辑“0”时，D<sub>0</sub> 触发器传递 1， $\overline{Q_0}$  输出 0，第一个译码器开始工作。

### 1、实验内容:

(1) **必做:** 用二进制编码设计一个“111001”序列检测器，对串行输入序列进行检测，当连续检测到 6 个码元符合检测码“111001”时，检测器输出为 1

**选做：**用 One-Hot 编码设计一个“111001”序列检测器，对串行输入序列进行检测，当连续检测到 6 个码元符合检测码“111001”时，检测器输出为 1 注意：“111001”序列的最后一个码元不能作为当前待测序列的第一个码元

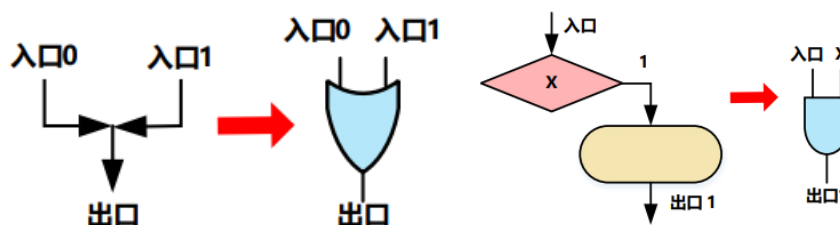
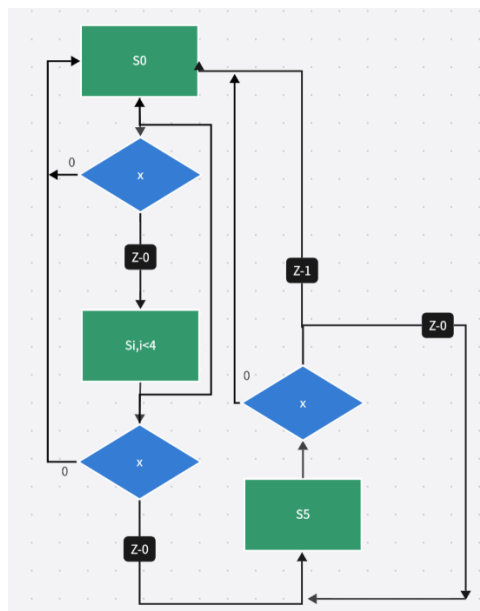
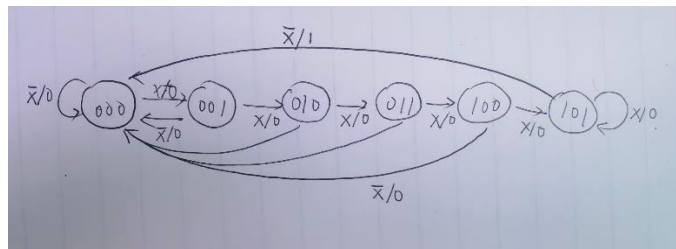
(2) 用静态（单步）的测试方法对实验结果进行验证

(3) 自行设计一个“1011011011100100”序列发生器用于对实验结果进行动态验证

(4) 用双踪示波器观察并记录动态验证结果。

## 2、实验要求：

(1) 根据设计要求划分设计层次、单元模块和接口信号，在报告上记录设计过程，绘制系统框图，每个模块的状态转移图和 ASM 图，并设计验证方案。



X 的判断是通过真值表写出逻辑函数式实现的。

(2) 用原理图输入法设计所有单元模块并编译，分析编译时产生的错误和警告信息

(3) 对所有的单元模块进行功能仿真，并记录和分析全部仿真结果

(4) 在顶层文件中连接全部单元模块并编译、综合、分配管脚和适配。

(5) 对整个系统进行时序仿真，并记录和分析仿真结果。

(6) 将仿真正确的设计下载到实验箱上，连接输入输出设备和示波器进行板级验证

1. 状态图

该工程包含的模块如下：

- (1) 待测信号产生模块。本质是序列发生器，包括手动输入模块与预置信号(1011011011100100)模块。
- (2) 判别相等模块，判断每一位信号是否相等。
- (3) 预定信号模块，即“111001”。本质是序列发生器。
- (4) 序列检测器模块。

判别相等：

判别数值是否相等的功能可以通过同或逻辑实现。由于器件中没有同或门，采用异或+非的逻辑实现。将 OD 与 Y 异或后得到 X，即判断两信号是否相等。

表 1 判断相等信号

OD	Y	X
0	0	1
1	1	
1	0	0
0	1	

序列检测器(Sequential Detector，SD)：

6 位 SD(二进制实现)：

表 2 二进制码实现 6 位序列检测器

文字描述				二进制编码						74161 集成计数器						
现态	次态	条件	输出	现态			次态			功能	端口					
				Q2	Q1	Q0	Q2	Q1	Q0		EN	LD	D3	D2	D1	D0
S0	S0	$\overline{X}$	0	0	0	0	0	0	0	保持	0	1	$\varphi$	$\varphi$	$\varphi$	$\varphi$
	S1	X	0				0	0	1	计数	1	1	$\varphi$	$\varphi$	$\varphi$	$\varphi$
S1	S0	$\overline{X}$	0	0	0	1	0	0	0	置数	0	0	0	0	0	0
	S2	X	0				0	1	0	计数	1	1	$\varphi$	$\varphi$	$\varphi$	$\varphi$
S2	S0	$\overline{X}$	0	0	1	0	0	0	0	置数	0	0	0	0	0	0
	S3	X	0				0	1	1	计数	1	1	$\varphi$	$\varphi$	$\varphi$	$\varphi$
S3	S0	$\overline{X}$	0	0	1	1	0	0	0	置数	0	0	0	0	0	0
	S4	X	0				1	0	0	计数	1	1	$\varphi$	$\varphi$	$\varphi$	$\varphi$
S4	S0	$\overline{X}$	0	1	0	0	0	0	0	置数	0	0	0	0	0	0
	S5	X	0				1	0	1	计数	1	1	$\varphi$	$\varphi$	$\varphi$	$\varphi$
S5	S0	$\overline{X}$	1	1	0	1	0	0	0	置数	0	0	0	0	0	0
	S5	X	0				1	1	0	保持	0	1	$\varphi$	$\varphi$	$\varphi$	$\varphi$

6 位 SD(One-Hot 码实现)：

表 3 One-Hot 码实现 6 位序列检测器

文字描述				One-Hot 码			
现	次	条	输	现态		次态	

态	态	件	出	Q5	Q4	Q3	Q2	Q1	Q0	Q5	Q4	Q3	Q2	Q1	Q0
S0	S0	$\bar{X}$	0	0	0	0	0	0	1	0	0	0	0	0	1
	S1	X	0							0	0	0	0	1	0
S1	S0	$\bar{X}$	0	0	0	0	0	1	0	0	0	0	0	0	1
	S2	X	0							0	0	0	1	0	0
S2	S0	$\bar{X}$	0	0	0	0	1	0	0	0	0	0	0	0	1
	S3	X	0							0	0	1	0	0	0
S3	S0	$\bar{X}$	0	0	0	1	0	0	0	0	0	0	0	0	1
	S4	X	0							0	1	0	0	0	0
S4	S0	$\bar{X}$	0	0	1	0	0	0	0	0	0	0	0	0	1
	S5	X	0							1	0	0	0	0	0
S5	S0	$\bar{X}$	1	1	0	0	0	0	0	0	0	0	0	0	1
	S5	X	0							1	0	0	0	0	0

### 待测信号：

由于手动输入待测信号逻辑简易，在这里给出用序列发生器产生“1011011100100”的真值表。用 74161+2 片 74138 的组合获得 16 位序列发生器。2 片 74138 组装为 4 线-16 线译码器，将对应地址与非输出即可得到待测信号“1011011011100100”。由于 74161 恰为 16 进制计数器，故无需特意改装，正常计数即可。

表 4 待测信号 4 线-16 线译码器组装

信号	高位芯片				低位芯片			
QD	EN1	EN2A	EN2B	功能	EN1	EN2A	EN2B	功能
0	0	x	x	封锁	1	0	0	工作
1	1	0	0	工作	x	1	1	封锁

表 5 待测信号输出 Y

地址输出									信号输出
Y0	Y2	Y3	Y5	Y6	Y8	Y9	Y10	Y13	Y
0	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	0	1	1	1
1	1	1	1	1	1	1	0	1	1
1	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	0

### 预定信号：

用 74161+74138 的组合获得 6 位序列发生器，原理同上。其中，由于是将预定信号与待测信号逐位比较，应在比较结果为“不相同”时重新从预定序列的开端开始比较。即，假如比较序列结果为“假”，预定信号应直接重新产生“111001”的序列。因此，置数端 LD 的接线需要特别说明。

表 6 预定信号输出 OD

地址输出				信号输出
Y0	Y1	Y2	Y5	OD
0	1	1	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

**预定序列 ASPRG 模块的 LD 端：**

由于 X 信号已经代表了当前 Y 与 OD 是否相等，且 X 输出为“0”时代表判断序列不相等，LD 端低电平有效。因此，X 可直接作为 LD 端的接入。考虑到预定序列的 ASPRG 本身是一个模 6 计数器，在从 0 计数到 5 后应当返回 0 计数，这两个条件是“或”的关系。即：序列不相等，或序列完整输出了，则重新计数。因此，将二者(X 与 $\bar{Y}_5$ )作与运算即可。

## 2. 状态转移方程

根据状态转移表求得状态转移方程如下。

**2.1 SD：**

SD(二进制码)：

$$EN = X \cdot \left( \sum_{i=0}^4 m_i^{n-1} \right)$$
$$LD = X + \bar{X} \cdot m_0^{n-1}$$
$$Z = \bar{X} \cdot m_5^{n-1}$$

其中， $m_i^{n-1}$ 代表（Q<sub>2</sub>Q<sub>1</sub>Q<sub>0</sub>）组成的最小项的现态。

SD(One-Hot 码)：

$$EN = X \cdot \left( \sum_{i=0}^4 m_i^{n-1} \right)$$
$$LD = X + \bar{X} \cdot m_0^{n-1}$$
$$Z = \bar{X} \cdot m_5^{n-1}$$

**2.2 ASPRG：**

自动加载数据“101...”时，用于产生待检测信号的 16 位 ASPRG 中，138 组装 4 线-16 线译码器，高位芯片与低位芯片的使能端分别如方程组（1）（2）所示：

$$\begin{cases} EN_2A = EN_2B = 0 \\ EN_1 = Q_D \end{cases} \quad (1)$$

$$\begin{cases} EN_2A = EN_2B = Q_D \\ EN_1 = 1 \end{cases} \quad (2)$$

而手动加载数据时，待测信号的输出仍用 Y 指代，则有：

$$Y = Input$$

**2.3 各组件连接部分**

由上文分析，对预定序列序列发生器(二进制码)LD 脚(记为 LDasprg)，有：

$$LDasprg = LD \cdot \bar{Y}_5$$

其中,  $\bar{Y}_5$  是该序列发生器 74138 的  $\bar{Y}_5$  脚输出。

同样, 对预定序列序列发生器(One-Hot 码)LD 脚, 有:

$$LDasprg = \overline{Nand} \cdot \bar{Y}_5$$

产生 X 的异或逻辑为:

$$X = Y \oplus OD$$

对于待测信号, 记 Model=0 时的 Y 为  $Y_0$ , Model=1 时的 Y 为  $Y_1$ , 则总电路的待测信号应有:

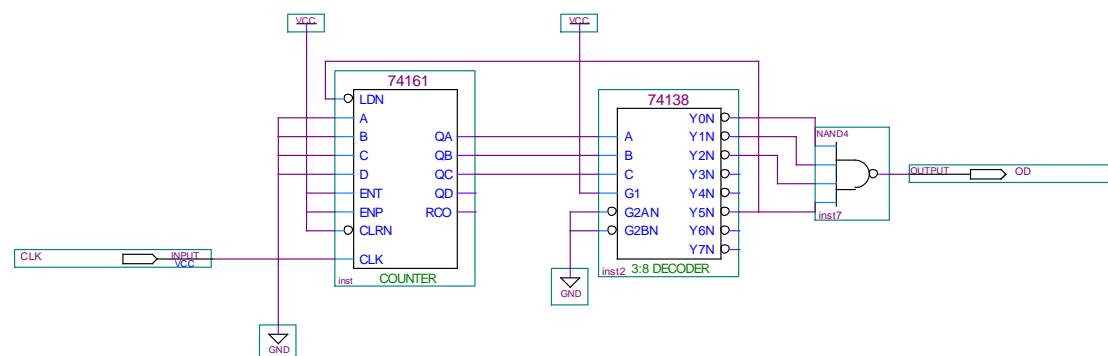
$$Y = Y_0 \cdot \overline{Model} + Y_1 \cdot Model$$

实际上, 由于手动输入的信号在 FPGA 板上是通过拨码开关代替的, 考虑到直接同上式接线会导致 Input 无视 CLK 信号, 在 CLK 低电平期间改变拨码开关状态同样会直接导致 Y 的改变。因此, 使 Input 经过了一个 D 锁存器的寄存, 锁存器的时钟信号与 CLK 同步。(验证表明, 这是必不可少的!! 因为没按 CLK 的时候换 Input 开关状态也计入信号检测, 直接导致信号判断为不相等!)

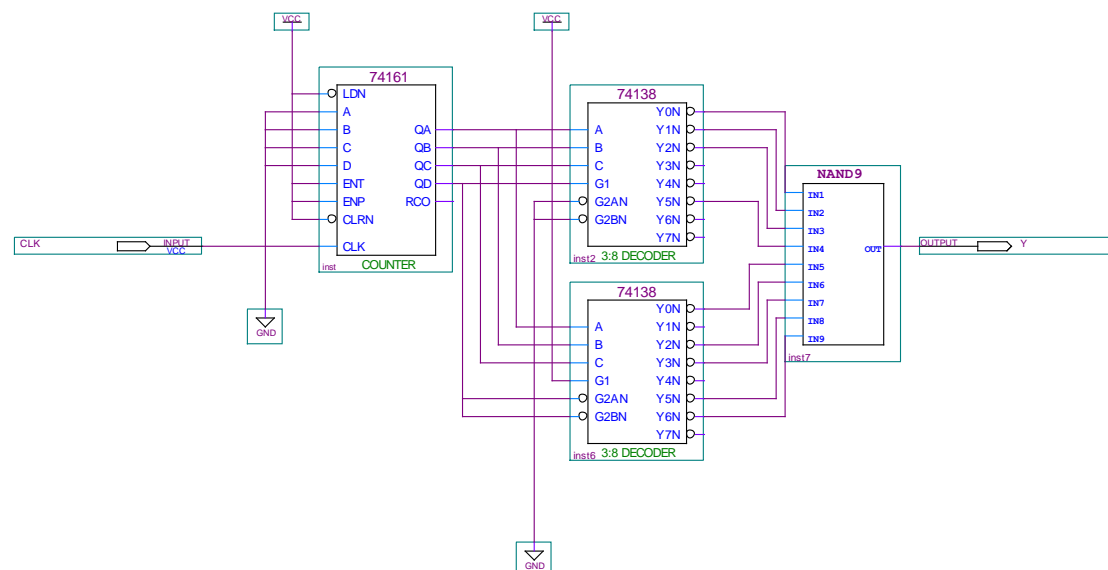
### 3. 电路原理图

首先给出 Model=0 (预置信号模式) 的电路图。

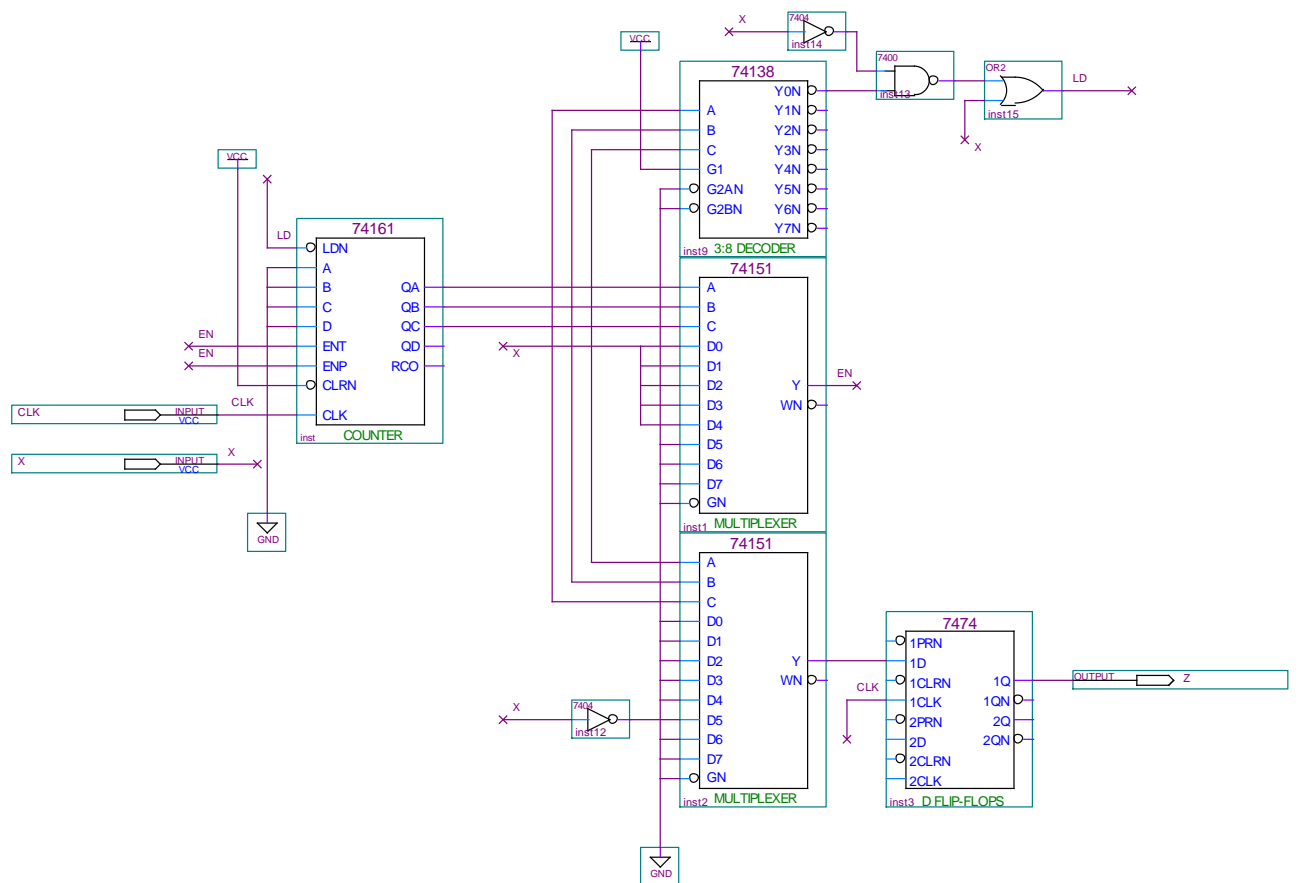
**预定序列序列发生器:**



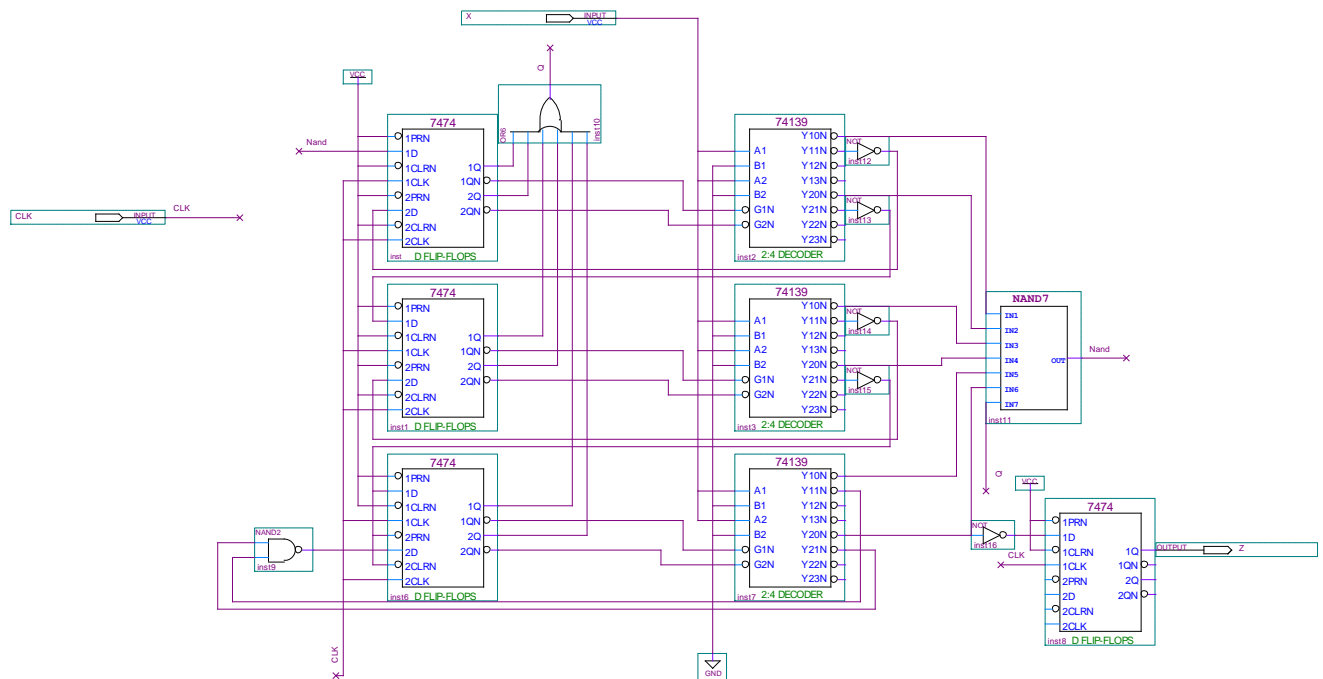
**待测信号序列发生器:**



**SD(二进制码实现):**

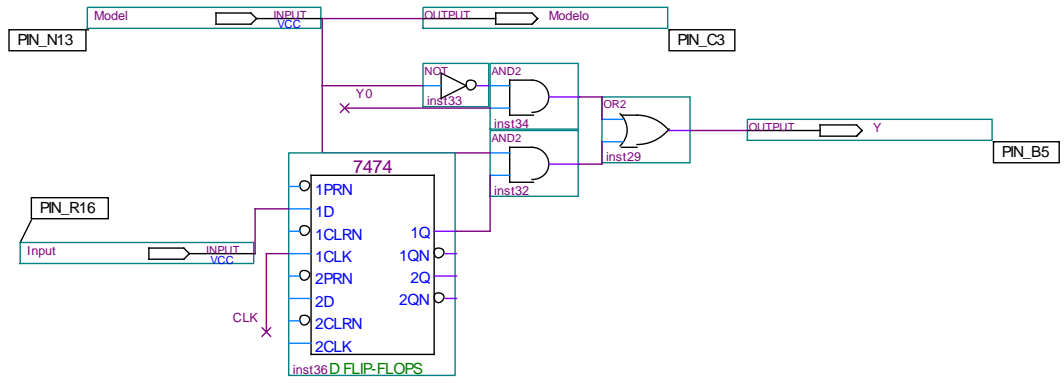


**SD(One-Hot 码实现):**

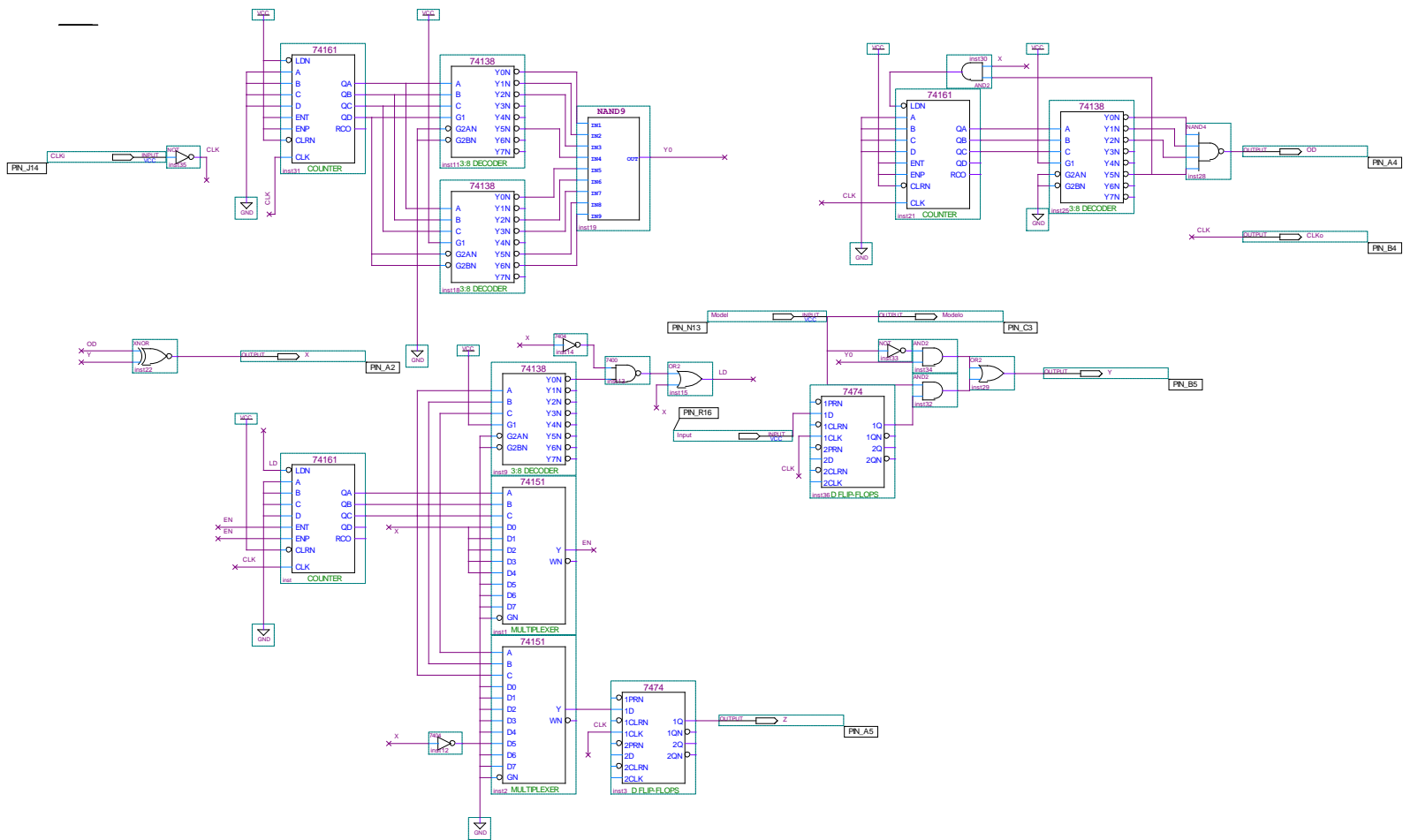


**待测信号产生合块(动态+手动):**

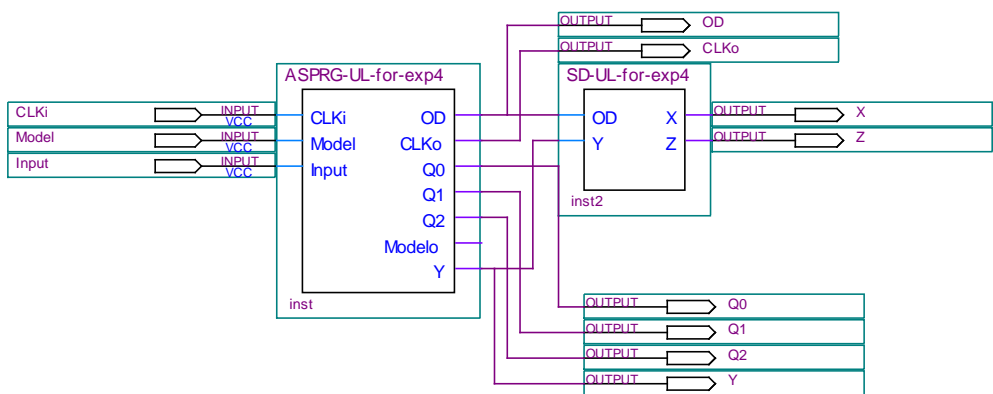




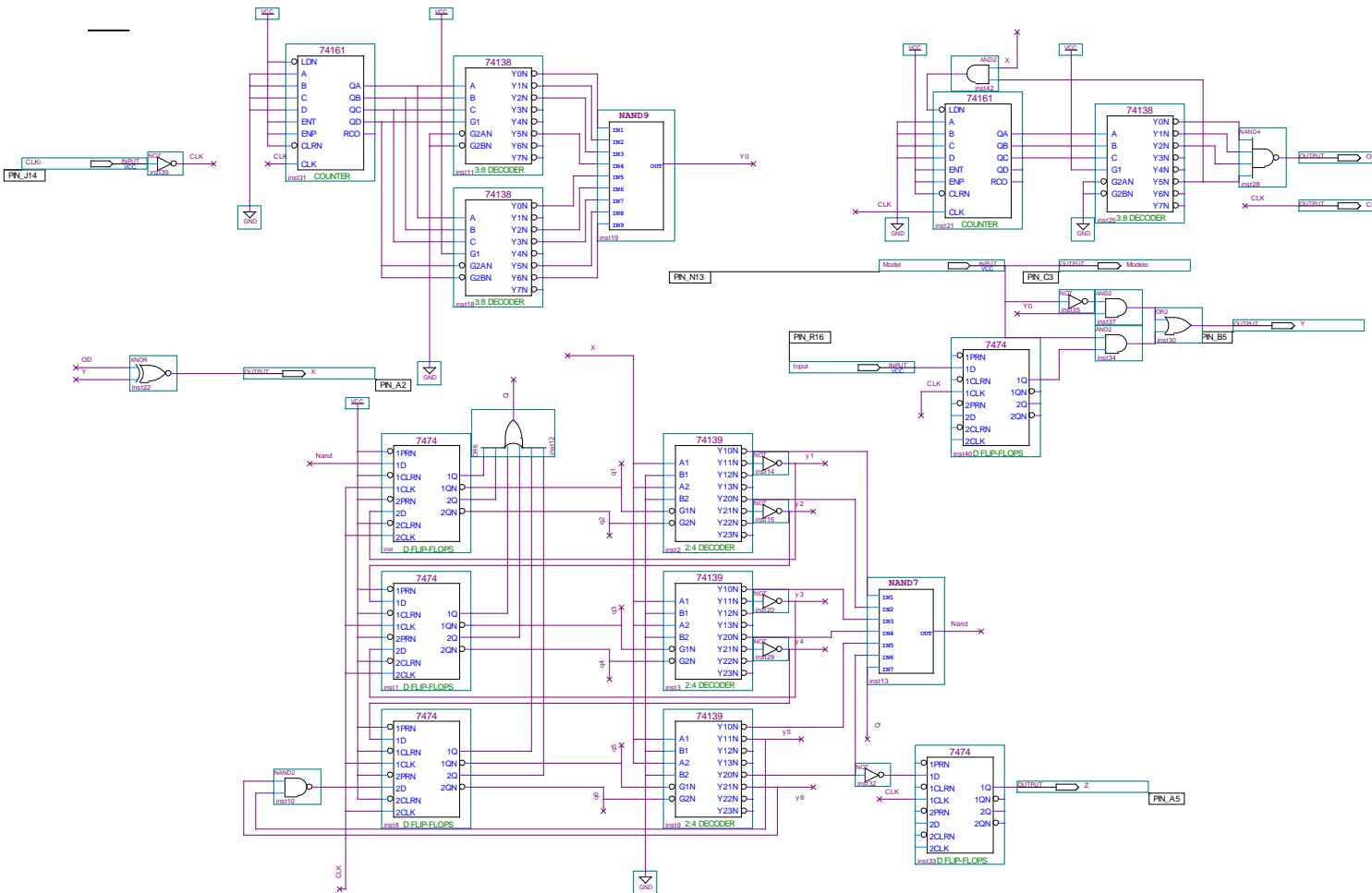
综合可得，总电路如下：  
总电路(二进制码实现)：



封装后的二进制码电路如下：



总电路(One-Hot 码实现):



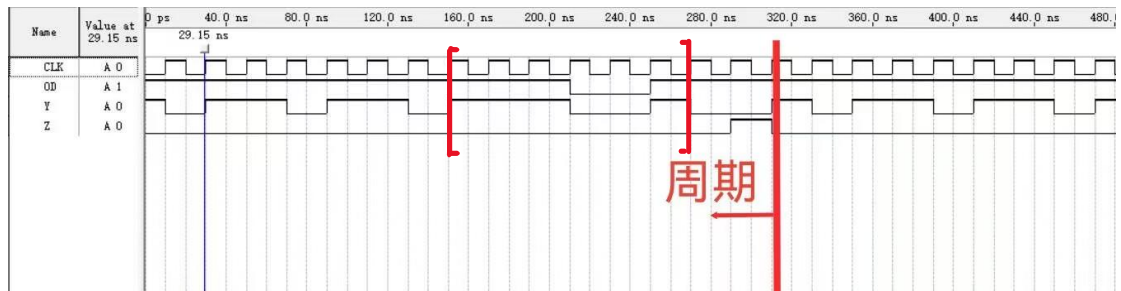
(由于 One-Hot 码电路未要求模块化，故未模块化实现)

## 4. 仿真

在此说明，仿真均是利用 Model=0 时的功能仿真。方便起见，直接采用了未合并的电路（即为加入 Model 与 Input 信号的电路）进行仿真。

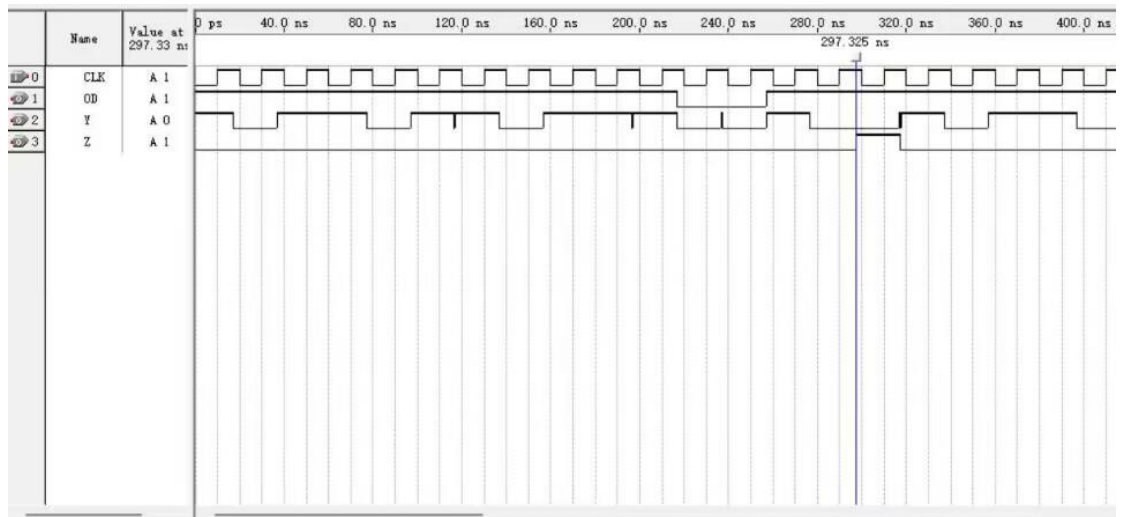
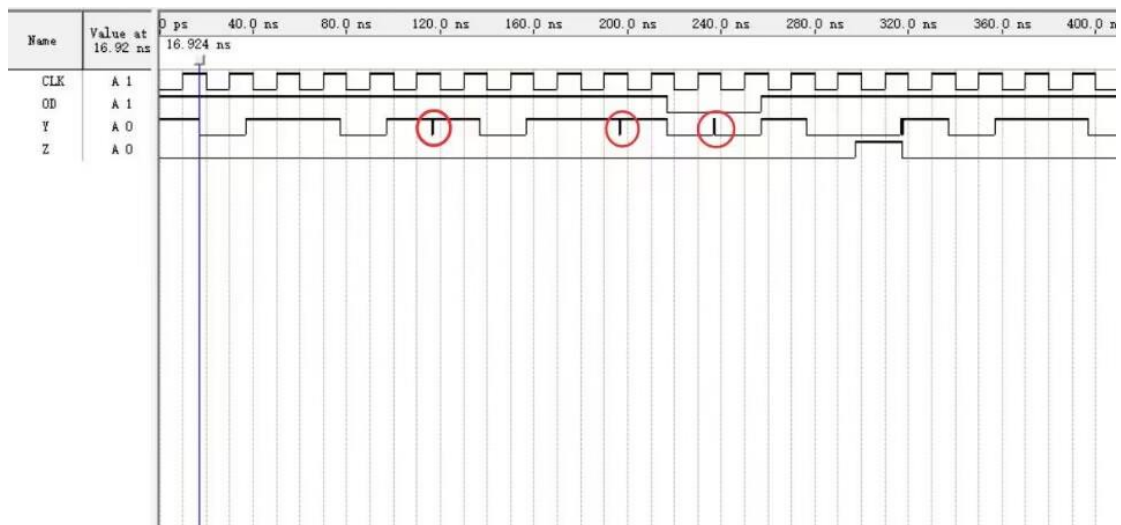
### 4.1 二进制码实现

功能仿真：



可以观察到，在相同波形（111001）后的一个周期后，Z 跃变为 1。延迟了两个周期分别是由输出端的 D 触发器、信号输入端自行设计的 D 触发器引起的。实验中并没有要求综合两种功能，所以如果不附加功能，应是一个周期的延迟。功能仿真正确。

### 时序仿真：



可以观察到，Y 信号约延迟 CLK 信号 6.924ns，Z 信号约延迟 CLK 信号 7.325ns。OD、Z 均无毛刺产生，毛刺主要集中在 Y 的波形上。以第一个毛刺为例，是在 138 低位芯片从 Y5 转到 Y6 时形成的，而此时高位芯片的 Y6 脚为低。观察发现毛刺出现时，不工作的芯片上均有工作芯片对应脚相反的信号。故推测毛刺的形成是两片 138 芯片之间使能端间不同步，产生竞争-冒险导致的。也因而，同样用于序列发生器的 OD 波形上没有毛刺产生。

同时，对照延迟信息：

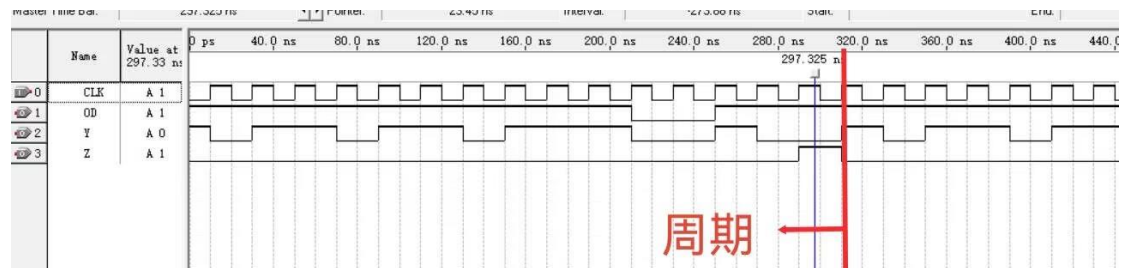
Clock to Output Times							
	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference	
1	OD	CLK	7.107	7.118	Rise	CLK	
2	Y	CLK	7.078	6.954	Rise	CLK	
3	Z	CLK	7.312	7.407	Rise	CLK	

发现均与全部毛刺的产生时间无法对应。因此可以排除 CLK 产生毛刺的假设。

同样可以发现，毛刺的产生对 Z 的判断没有影响。

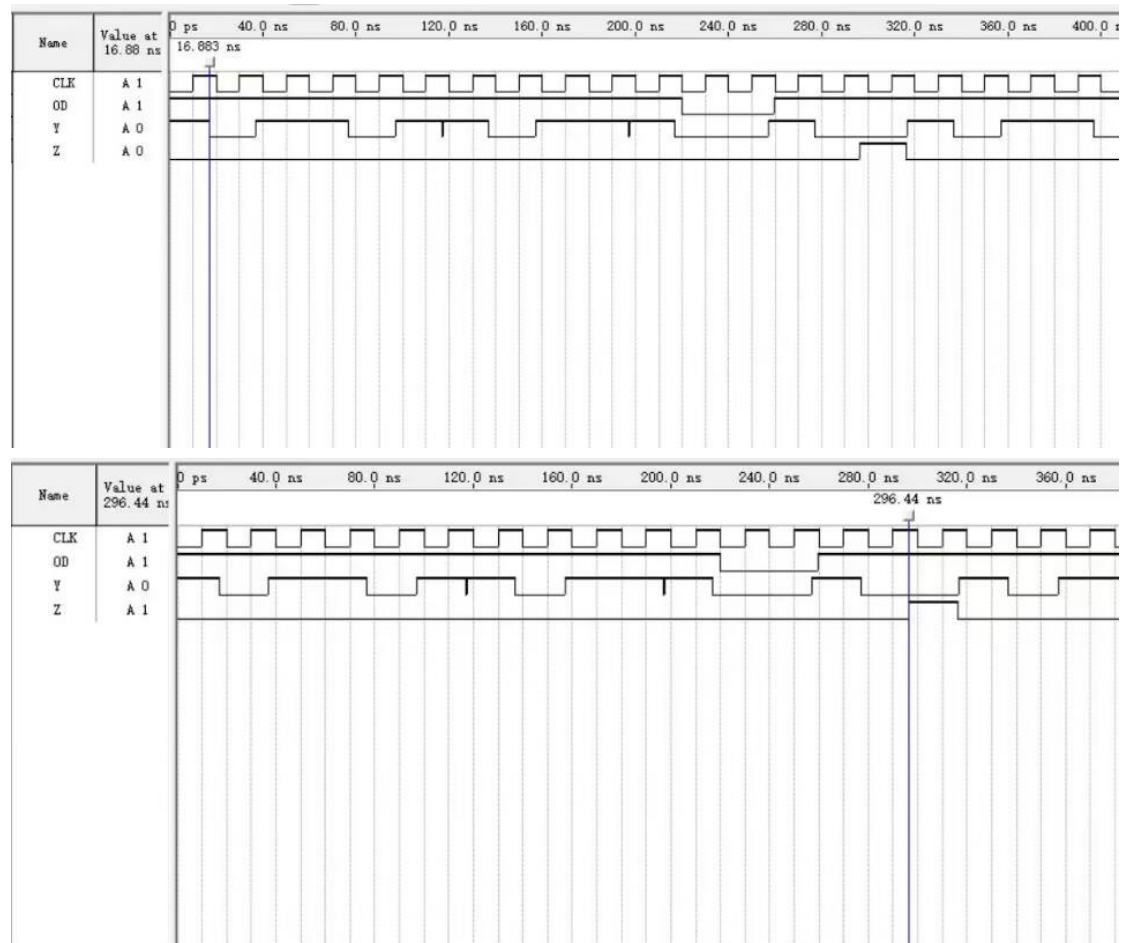
## 4.2 One-Hot 码实现

功能仿真：



分析同二进制码电路，功能正确。

时序仿真：



可以观察到，Y 信号约延迟 CLK 信号 6.883ns，Z 信号约延迟 CLK 信号 6.44ns。毛刺的分

析同二进制码电路。

5. 管脚分配

管脚分配如下。其中，为便于实现静态验证，CLK 接到按键 J14 上。

<div><div><div></div><div>&lt;&lt;&gt;&gt;</div></div>Edit: <div><div></div><div></div></div>PIN_C3</div>						
Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved
<div><div></div>CLK</div>	Input	PIN_J14	5	B5_N0	2.5 V (default)	
<div><div></div>CLKo</div>	Output	PIN_B4	8	B8_N0	2.5 V (default)	
<div><div></div>Input</div>	Input	PIN_R16	5	B5_N0	2.5 V (default)	
<div><div></div>Model</div>	Input	PIN_N13	5	B5_N0	2.5 V (default)	
<div><div></div>Modelo</div>	Output	PIN_C3	8	B8_N0	2.5 V (default)	
<div><div></div>OD</div>	Output	PIN_A4	8	B8_N0	2.5 V (default)	
<div><div></div>Y</div>	Output	PIN_B5	8	B8_N0	2.5 V (default)	
<div><div></div>Z</div>	Output	PIN_A5	8	B8_N0	2.5 V (default)	
<new node>>						

6. 验证方案

预置序列验证（1011011011100100）时，仅需将开关 SW7 断开，按动按钮 KEY0 产生单次静态脉冲作为 CLK 验证。手动输入序列验证时，将开关 SW7 闭合，同时改变开关 SW0 的状态后按动按钮产生单次静态脉冲。SW0 断开时按动按钮等效于输入“0”，闭合时按动按钮等效于输入“1”。

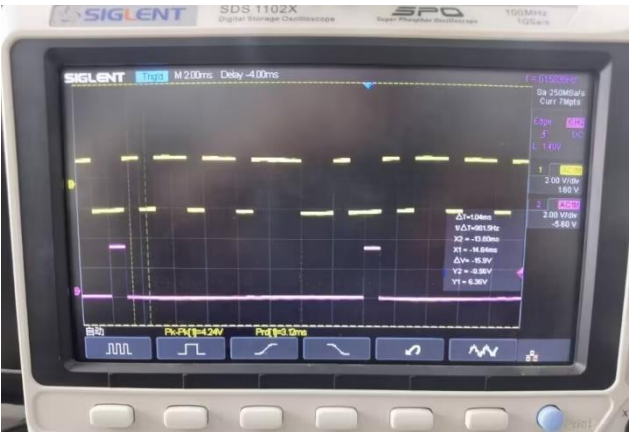
手动输入序列验证时，输入不同的序列信号，观察是否能正常实现功能。

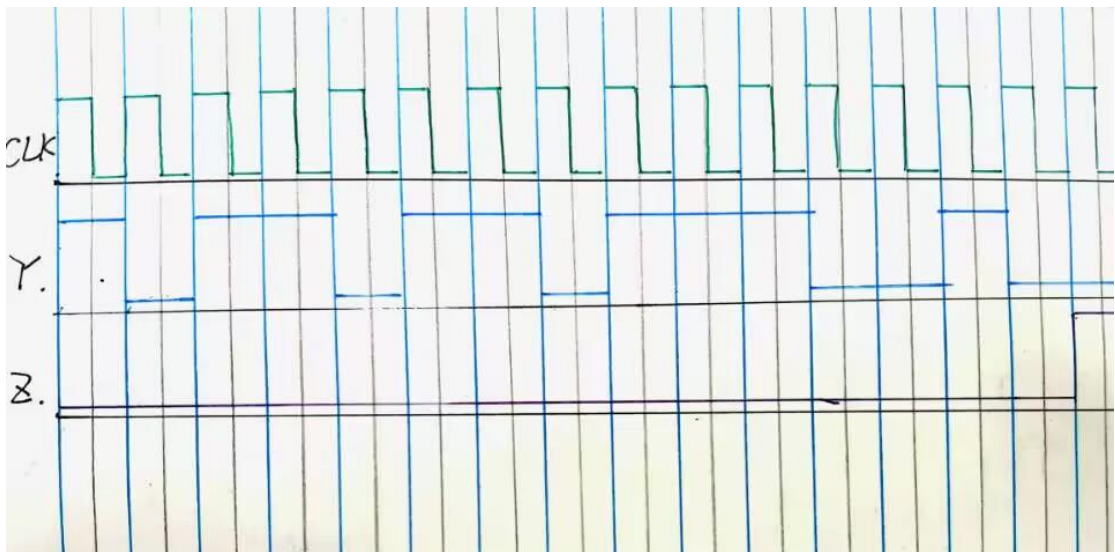
三、实验记录

（记录实验具体步骤、原始数据、实验过程、实验中遇到的故障现象、排除故障的过程和方法等）

**实验步骤：**首先利用脉冲按钮进行静态验证。静态验证时，输入信号的获取同上文。改变输入信号后按下按钮，信号输入不正确时，计数器跳转回 0。循环输入正确信号“0 111001 0 111001”时，代表检测正确的 LED 灯能够正确亮灯。静态验证结果符合功能。

再利用双踪示波器进行动态验证。利用信号源产生频率 1kHz、高电平 5V、低电平 0V 的方波作为 CLK 信号，将功能调整至 Model=0（预置信号），显示(CLK,Y)、(Y,Z)波形，记录结果。





波形说明，连续 CLK 信号输入时，Y 循环输出“1011011011100100”信号，Z 在延迟后输出跃变为“1”。动态验证结果符合功能。

经检验，该电路能够自启动（输入信号错误时计数器归零）。

**遇到的问题及解决方法：**同上文，实验中无问题。

## 四、实验仪器

FPGA 板，易派箱

## 五、实验小结 （总结实验完成情况，对设计方案和实验结果做必要的讨论，简述实验收获和体会）

本次实验周期较长，从验收的两周前就开始了。也因此，这次实验所设计的功能较为完善。

这次实验中遇到的一个困难是自行输入信号进行验证。

首先，是输入信号与 CLK 信号同步的问题。虽然如上文所说，利用一个 D 锁存器解决了这个困难，但实际上在自行验证的时候，错误的现象仅仅是输出信号几乎一直不变，完全想不到是哪里出错了。最后错误的排查也不是通过实验排查的，是检查电路时，灵光一闪，突然感觉到这里有问题才查出来的。事实上，这个错误是可以通过实验排查的，因为在并没有动 CLK 按钮时，拨动拨码开关，计数器会直接清零。这说明，连出计数器很重要！

其次，自行验证时设计验证方案也是一个困难。本次实验对全方位均有较高要求，对设计电路的能力也有较大提升。

## 六、参考资料

## 《数字逻辑设计实践 2023 年教学计划 A\_V1.0》